

Lossless Video for Recording and Archiving

An article by Space Banana

The wonder of lossless compression lets us preserve our information while using less space. On the video space, we have a significant amount of encoding formats that support lossless encoding, as well as encoders that take advantage of this. So, what should we use? What solution is heavier on the CPU? What encoder produces smaller files? What's more lightweight to playback? In this article, we will find out.

What's "lossless"?

In digital compression, we have 3 main types: uncompressed, lossy and lossless.

Uncompressed is straightforward, it means there is no data compression in the encode.

Lossy is a compression algorithm that loses and destroys information, to produce significantly small file sizes.

Lossless is a compression algorithm that lowers the digital size of a piece of information, without destroying or losing any information.

Lossless Solutions in Video

Most people produce lossy video material all the time, even professionals! For consumption, delivery and the web, nowadays the vast majority of people encode their videos in the H.264 format, with VP9, AV1 and H.265 slowly gaining popularity. People handling professional video usually handle videos encoded in H.264, ProRes, DnxHD/DNxHR and Cineform.

Out of all these mentioned formats, only H.264, H.265, VP9 and AV1 support lossless encoding. But are these our only available solutions for lossless encoding? Of course not! We also have encoders that implement huffman-based formats, such as HuffYUV, Utvideo and MagicYUV. We can also use image formats in video, such as PNG, TIFF, etc.

Lossless Encoding for Footage and Archiving

When recording and archiving video material, regardless of its purpose, we need to keep in mind a few technical aspects.

If we want to perfectly preserve all pixel information the way it originally is, we need to consider not only lossless compression but also how we encode color. Your lossless encoder should support RGB, as well as higher bit depths if your source video has a high bit depth. Does your source have transparency? Then

we also need an alpha channel. In case of not having RGB support, does the YUV implementation your encoder uses support 4:4:4 sampling?

Considering video recording, your lossless codec should be fast and lightweight to encode, so you can make use of it without needing to get yourself a very fast and proportionally expensive CPU. Generally, faster encode configurations produce bigger file sizes, so you have to find your ideal tradeoff.

Are you tight on space and is your network speed slow? Or do you have terabytes of disk space to use and you have blazingly fast internet connectivity? The bitrate (and consequentially the file size) of your video will matter to you. If you have network speed and space concerns, you should aim for an encoding solution that produces smaller files.

If you are going to use your video material on video editing, VFX, or similar tasks, you will want to use an encoder and format that are lightweight to decode. The faster it is to decode, the higher your performance will be in video editing. The complexity of the compression algorithm is inversely proportional to the file size of your video. Intraframe compression is also easier to decode, but can result in much larger files.

Intra-frame and Inter-frame

Interframe and intraframe are temporal compression methods. Intraframe compression has no temporal compression, this means that the compression of one video frame does not affect the compression of any other. Each frame uses unique data for all pixels, producing bigger file sizes but making your video much faster and more lightweight to decode.

Interframe acts the other way. This compression method uses temporal compression to share data between frames. You have keyframes, which use fully unique data, and between these keyframes you have frames that share identical pixel data with each other. If, until the next keyframe, there's a pixel that doesn't change between frames, that data will be shared between those frames until the pixel changes or the next keyframe is reached.

Interframe compression produces significantly smaller video files if their content is static or has static parts. It is however, harder to decode an interframe video, and that matters a lot for performance in video editing, conversion and other tasks.

Encoder Benchmarks

I will compare the different characteristics of encoders and encoding formats in the context of recording and encoding lossless video. All tests are done with FFmpeg.

I have selected the following encoders:

- libx264rgb
- utvideo
- png

For x264, I will perform multiple test variants:

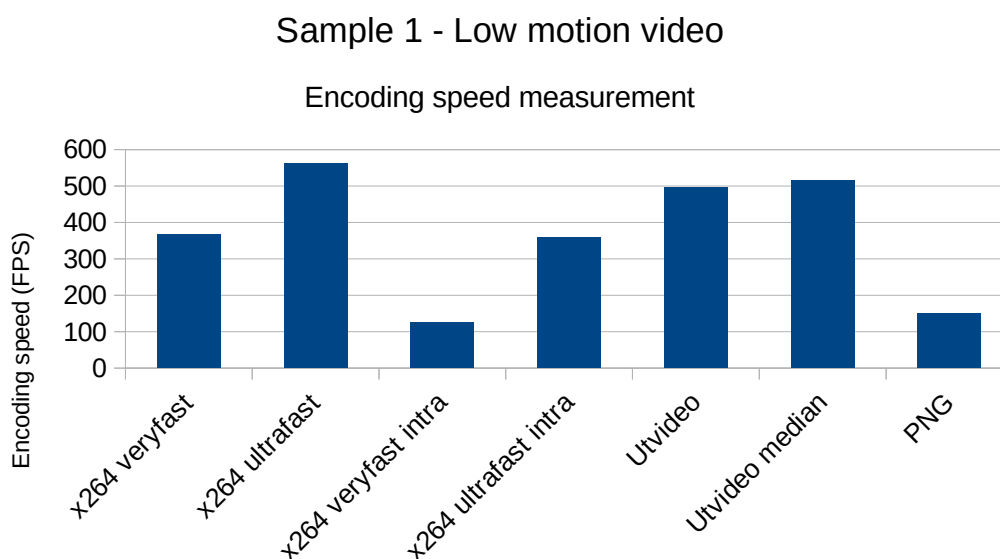
- veryfast preset, interframe
- veryfast preset, intraframe
- ultrafast preset, interframe
- ultrafast preset, intraframe

For Utvideo, I will perform 2 tests, one with the default prediction method (left), and the other with the median prediction method.

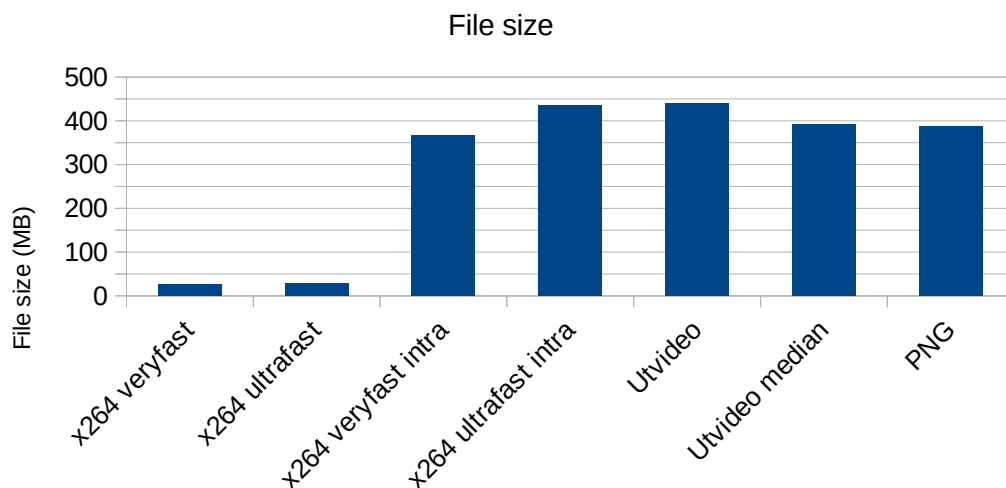
I won't test VP9, H.256 and AV1 because these formats, despite supporting lossless encoding, are very heavy to encode. Because of this, they are not reliable for recording, and not as reliable for archiving.

For the test, a lossless RGB video is transcoded into all variants, and encoding speed and file size are measured. Each sample is 20 seconds long. Source video is a 640x480 60 FPS video. All tests are done twice, for 2 different video samples. Sample 1 has low motion and sample 2 has high motion. This is important for observing when interframe compression is ideal.

Here are the test results:



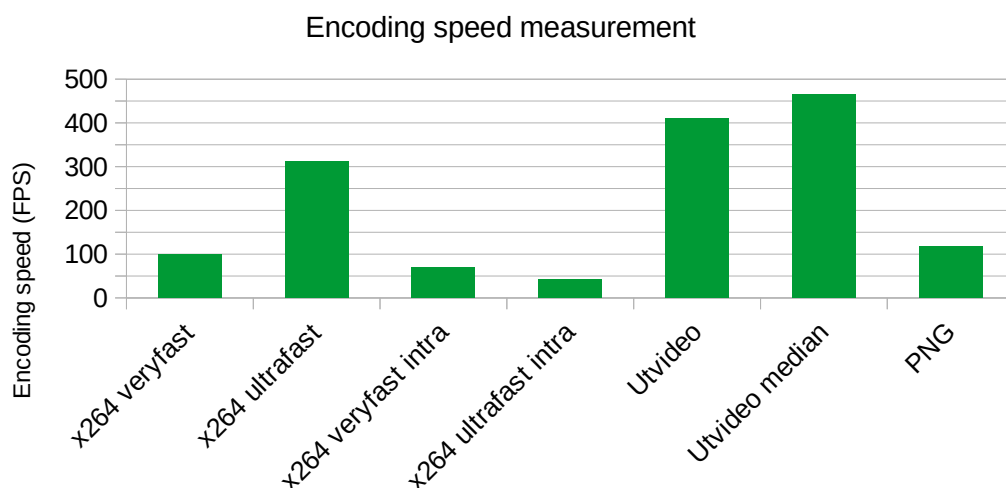
Sample 1 - Low motion video



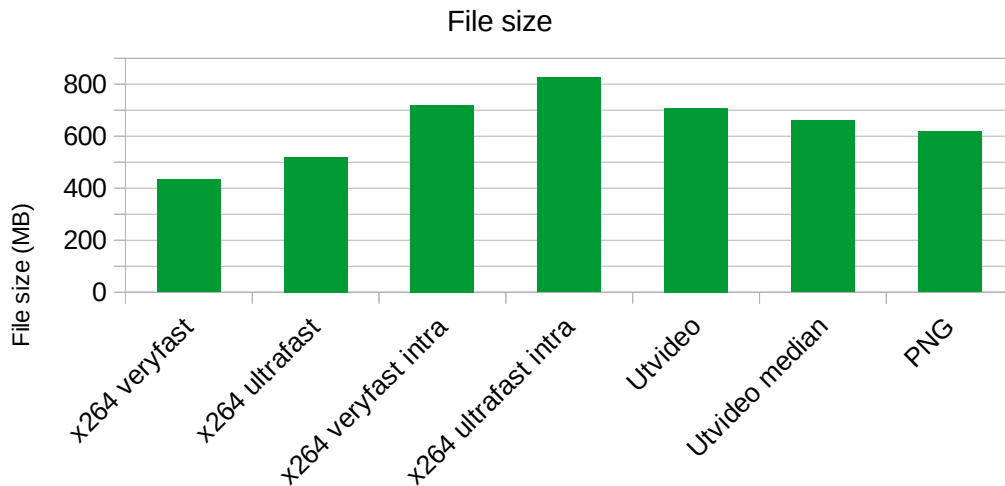
The results are very interesting. As for the encoding speed, x264 with the ultrafast preset and Utvideo win. Utvideo median is a better prediction algorithm, and surprisingly it doesn't lower encoding speed. Weirdly enough, using intraframe compression on x264 significantly worsens the encoding speed.

In the file size graph, we can see the advantage of using interframe encoding. Sample 1 is a low-motion sample scene, and so using interframe compression significantly improves file sizes. Intraframe solutions here have more or less the same file size, and so whichever is fastest and has the feature set you need wins.

Sample 2 - High motion video



Sample 2 - High motion video



Encoding speed lowered on all tests in Sample 2, probably due to having less pixels with the same color due to the high and complex motion. Utvideo still holds the place of the fastest encoder.

The file sizes are now much more similar between each other than in the previous sample. Since sample 2 has more motion, the advantage of using interframe compression is lower, and so the regular x264 tests are much closer to the intraframe x264 tests. Remember that interframe compression is slower to decode.

Conclusion

Utvideo is a very competitive codec for lossless recording and archiving. It delivers great performance both on the encoding and decoding sides and it has extensive color format support. Utvideo's main downside is the file size, and that's where slower, more complex compression algorithms with interframe compression enter. The x264 encoder is ideal for producing lossless video with less footprint, although the encoding speed is a bit slower. Using the ultrafast preset approaches x264 to Utvideo's encoding performance, at the cost of producing file sizes much closer to Utvideo.

x264 in intraframe compression somehow has a significantly lower encoding performance, so if you are looking for intraframe compression, make sure to look for the other solutions.

PNG isn't ideal for video encoding. It's slow to encode, slow to decode and still produces large files.